PORTAL
USPTO

Subscribe (Full Service) Register (Limited Service, Free) Login

Search: The ACM Digital Library The Guide

"Decision Trees and Diagrams" + "sifting variables"

THE ACM DIGITAL LIBRARY

Feedback Report a problem Satisfaction survey

Terms used and **Decision Trees Diagrams sifting variables** Found **8** of **171,143**

Sort results by relevance

Display results expanded form

Save results to a Binder

Search Tips

Open results in a new window

Try an Advanced Search

Try this search in The ACM Guide

Results 1 - 8 of 8

Relevance scale

1 Decision Trees and Diagrams

Bernard M. E. Moret

December 1982 **ACM Computing Surveys (CSUR)**, Volume 14 Issue 4

Publisher: ACM Press

Full text available: pdf(2.68 MB) Additional Information: full citation, references, citings, index terms

2 Generating Decision Trees for Decoding Binaries

Henrik Theiling

August 2001 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN workshop on Languages, compilers and tools for embedded systems LCTES '01 , Proceedings of the 2001 ACM SIGPLAN workshop on Optimization of middleware and distributed systems OM '01**, Volume 36 Issue 8

Publisher: ACM Press

Full text available: pdf(208.24 KB) Additional Information: full citation, abstract, references, citings, index terms

Tools reading binary code, like analysers, debuggers, disassemblers, etc., need to decode the target's machine code. A decision tree is often used to represent the decoding function.

Manually writing a decoder is a lengthy and error-prone task. It is desirable to be able to use the vendor's instruction code manual and to easily transform the documentation into a specification that a tool can use to generate a decoder.

This paper presents a novel algorithm that computes a decis ...

3 A message-based fault diagnosis procedure

J R Agre

August 1986 **ACM SIGCOMM Computer Communication Review , Proceedings of the ACM SIGCOMM conference on Communications architectures & protocols SIGCOMM '86**, Volume 16 Issue 3

Publisher: ACM Press

Full text available: pdf(1.10 MB) Additional Information: full citation, abstract, references, citings, index terms

A new diagnostic message protocol that provides fault diagnosis capabilities for the

communications in a distributed system environment is described. The protocol is designed to operate in conjunction with a standard end-to-end communication protocol and uses special messages to determine the system fault state. A diagnosis message is represented using a test dependency model that is derived from the system topology. These messages are used by an adaptive strategy designed to achieve specif ...

4 Binary Decision Diagrams: From abstract representations to physical implementations

Jose S. Metos, John V. Oldfield

June 1983 **Proceedings of the 20th conference on Design automation**

Publisher: IEEE Press

Full text available: pdf(355.45 KB) Additional Information: full citation, abstract, references, index terms

The Binary Decision Diagram (BDD) has established its place as a modelling technique for digital systems. The paper shows that further significant benefits can be realized by implementing BDDs as custom or semi-custom integrated circuits. Benefits include efficient use of silicon area and improved simulation. A well-known example is given of a traffic light controller previously implemented by a PLA. The BDD layout is regular, compact and lends itself to automatic generation. A sample worst ...

5 Finding the optimal variable ordering for binary decision diagrams

S. J. Friedman, K. J. Supowit

October 1987 **Proceedings of the 24th ACM/IEEE conference on Design automation**

Publisher: ACM Press

Full text available: pdf(612.97 KB) Additional Information: full citation, abstract, references, citings, index terms

The ordered binary decision diagram is a canonical representation for Boolean functions, presented by Bryant as a compact representation for a broad class of interesting functions derived from circuits. However, the size of the diagram is very sensitive to the choice of ordering on the variables; hence for some applications, such as Differential Cascode Voltage Switch (DCVS) trees, it becomes extremely important to find the ordering leading to the most compact representation. We present an ...

6 Lazy group sifting for efficient symbolic state traversal of FSMs

Hiroyuki Higuchi, Fabio Somenzi

November 1999 **Proceedings of the 1999 IEEE/ACM international conference on Computer-aided design**

Publisher: IEEE Press

Full text available: pdf(108.64 KB) Additional Information: full citation, abstract, references, citings, index terms

This paper proposes lazy group sifting for dynamic variable re-ordering during state traversal. The proposed method relaxes the idea of pairwise grouping of present state variables and their corresponding next state variables. This is done to produce better variable orderings during image computation without causing BDD size blowup in the substitution of next state variables with present state variables at the end of image computation. Experimental results show that our approach is more rob ...

7 Proving circuit correctness using formal comparison between expected and extracted behaviour

Jean-Christophe Madre, Jean-Paul Billon

June 1988 **Proceedings of the 25th ACM/IEEE conference on Design automation**

Publisher: IEEE Computer Society Press

Full text available: pdf(852.70 KB) Additional Information: full citation, abstract, references, citings, index terms

This paper presents a new method for verifying functionality in the design of VLSI circuits. Our method fits naturally in a methodology based on a Hardware Description Language (HDL). Two programs describe the system under design: (1) its specification and (2) the extracted behaviour from its layout. Verifying the design comes down to proving that these programs are correct and equivalent with regard to the HDL semantics. We define a process named F ...

8 Prime rule-based methodologies give inadequate control

J R B Cockett, J Herrera

December 1986 **Proceedings of the ACM SIGART international symposium on Methodologies for intelligent systems**

Publisher: ACM Press

Full text available: pdf(590.81 KB) Additional Information: full citation, abstract, references, citings, index terms

The use of rule-based methodologies in the development of Expert Systems is widespread. In order to provide good explanations in these systems it is desirable that the rules be prime. The difficulty of expressing control in such rules, and thus arriving at a desirable sequencing of events, has led to pragmatic additions to the basic methodology. Recent developments in the theory of decision processes have provided new insight into the form of a desirable sequencing. Prime rules, even when a ...

Results 1 - 8 of 8

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2006 ACM, Inc.
Terms of Usage Privacy Policy Code of Ethics Contact Us

Useful downloads: Adobe Acrobat QuickTime Windows Media Player Real Player

PORTAL
USPTO

Subscribe (Full Service) Register (Limited Service, Free) Login

Search: ◉ The ACM Digital Library ○ The Guide

"Decision Trees and Diagrams" + "sifting variables" + "labels

THE ACM DIGITAL LIBRARY

Feedback Report a problem Satisfaction survey

Terms used and **Decision Trees Diagrams sifting variables labels restructure BDD**

Found **9** of **171,143**

Sort results by: relevance

Display results: expanded form

Save results to a Binder

Search Tips

☐ Open results in a new window

Try an Advanced Search

Try this search in The ACM Guide

Results 1 - 9 of 9

Relevance scale

1 Decision Trees and Diagrams

Bernard M. E. Moret

December 1982 **ACM Computing Surveys (CSUR)**, Volume 14 Issue 4

Publisher: ACM Press

Full text available: pdf(2.68 MB) Additional Information: full citation, references, citings, index terms

2 Generating Decision Trees for Decoding Binaries

Henrik Theiling

August 2001 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN workshop on Languages, compilers and tools for embedded systems LCTES '01 , Proceedings of the 2001 ACM SIGPLAN workshop on Optimization of middleware and distributed systems OM '01**, Volume 36 Issue 8

Publisher: ACM Press

Full text available: pdf(208.24 KB) Additional Information: full citation, abstract, references, citings, index terms

Tools reading binary code, like analysers, debuggers, disassemblers, etc., need to decode the target's machine code. A decision tree is often used to represent the decoding function.

Manually writing a decoder is a lengthy and error-prone task. It is desirable to be able to use the vendor's instruction code manual and to easily transform the documentation into a specification that a tool can use to generate a decoder.

This paper presents a novel algorithm that computes a decis ...

3 Decision trees for geometric models

Esther M. Arkin, Henk Meijer, Joseph S. B. Mitchell, David Rappaport, Steven S. Skiena

July 1993 **Proceedings of the ninth annual symposium on Computational geometry**

Publisher: ACM Press

Full text available: pdf(1.15 MB) Additional Information: full citation, abstract, references, citings, index terms

A fundamental problem in model-based computer vision is that of identifying which of a given set of geometric models is present in an image. Considering a "probe" to be an

oracle that tells us whether or not a model is present at a given point, we study the problem of computing efficient strategies ("decision trees") for probing an image, with the goal to minimize the number of probes necessary (in the worst case) to determine which single model is present. We show ...

4 Retargetable tools for embedded software: Automated synthesis of efficient binary decoders for retargetable software toolkits

Wei Qin, Sharad Malik

June 2003 **Proceedings of the 40th conference on Design automation**

Publisher: ACM Press

Full text available: pdf(152.35 KB) Additional Information: full citation, abstract, references, citings, index terms

A binary decoder is a common component of software development tools such as instruction set simulators, disassemblers and debuggers. The efficiency of the decoder can have a significant impact on the efficiency of these software tools. Automated synthesis of efficient binary decoders is therefore necessary for retargetable software tool development frameworks targeting the rapidly growing field of application-specific processor design. This paper describes a decoder synthesis algorithm that tra ...

Keywords: binary decoder, decision tree, decoding tree, instruction set simulator

5 A message-based fault diagnosis procedure

J R Agre

August 1986 **ACM SIGCOMM Computer Communication Review , Proceedings of the ACM SIGCOMM conference on Communications architectures & protocols SIGCOMM '86**, Volume 16 Issue 3

Publisher: ACM Press

Full text available: pdf(1.10 MB) Additional Information: full citation, abstract, references, citings, index terms

A new diagnostic message protocol that provides fault diagnosis capabilities for the communications in a distributed system environment is described. The protocol is designed to operate in conjunction with a standard end-to-end communication protocol and uses special messages to determine the system fault state. A diagnosis message is represented using a test dependency model that is derived from the system topology. These messages are used by an adaptive strategy designed to achieve specif ...

6 Finding the optimal variable ordering for binary decision diagrams

S. J. Friedman, K. J. Supowit

October 1987 **Proceedings of the 24th ACM/IEEE conference on Design automation**

Publisher: ACM Press

Full text available: pdf(612.97 KB) Additional Information: full citation, abstract, references, citings, index terms

The ordered binary decision diagram is a canonical representation for Boolean functions, presented by Bryant as a compact representation for a broad class of interesting functions derived from circuits. However, the size of the diagram is very sensitive to the choice of ordering on the variables; hence for some applications, such as Differential Cascode Voltage Switch (DCVS) trees, it becomes extremely important to find the ordering leading to the most compact representation. We present an ...

7 A comparison of techniques for the specification of external system behavior

Alan,M. Davis

August 1988 **Communications of the ACM**, Volume 31 Issue 9

Publisher: ACM Press

Full text available: pdf(2.95 MB) Additional Information: full citation, abstract, references, citings, index terms

The elimination of ambiguity, inconsistency, and incompleteness in a Software Requirements Specification (SRS) document is inherently difficult, due to the use of natural language. The focus here is a survey of available techniques designed to reduce these negatives in the documentation of a software product's external behavior.

8 The multiple variable order problem for binary decision diagrams: theory and practical application

Christoph Scholl, Bernd Becker, Andreas Brogle

January 2001 **Proceedings of the 2001 conference on Asia South Pacific design automation**

Publisher: ACM Press

Full text available: pdf(134.00 KB) Additional Information: full citation, abstract, references, index terms

Reduced Ordered Binary Decision Diagrams (ROBDDs) gained widespread use in logic design verification, test generation, fault simulation, and logic synthesis [17, 7]. Since the size of an ROBDD heavily depends on the variable order used, there is a strong need to find variable orders that minimize the number of nodes in an ROBDD. In certain applications we have to cope with ROBDDs with different variable orders, whereas further manipulations of these ROBDDs require common variable orders. In ...

9 Decision tree reduction

J. R. B. Cockett, J. A. Herrera

October 1990 **Journal of the ACM (JACM)**, Volume 37 Issue 4

Publisher: ACM Press

Full text available: pdf(2.08 MB) Additional Information: full citation, abstract, references, citings, index terms

The reduction algorithm is a technique for improving a decision tree in the abseence of aproecise cost criterion. The result of applying the algorithm is an irreducible tree that is no less efficient than the original, and may be more efficient. Irreducible trees arise in discrete decision theory as an algebraic form for decision trees. This form has significant computational properties. In fact, every irreducible is optimal with respect to some expected testing cost criterion and is stric ...

Results 1 - 9 of 9

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2006 ACM, Inc.
Terms of Usage Privacy Policy Code of Ethics Contact Us

Useful downloads: Adobe Acrobat QuickTime Windows Media Player Real Player

	Type	L #	Hits	Search Text
1	BRS	L1	0	((binary adj decision adj diagram) or (BDD)) same restructure
2	BRS	L2	0	((binary adj decision adj diagram) or (BDD)) near restructure
3	BRS	L3	5	((binary adj decision adj diagram) or (BDD)) and restructure
4	BRS	L4	0	(restructure same sift$3) and (hardware same binary)
5	BRS	L5	570	716/16.ccls.
6	BRS	L6	1698	703/2.ccls.
7	BRS	L7	2	703/2.ccls. and BDD and (tree$2 same leaves) and (nodes same graph)
8	BRS	L8	4	((variable$2) same BDD) and (first adj storage) and (second adj storage)
9	BRS	L9	0	((variable$2) same BDD) and (first adj storage) and (second adj storage) and leave$2 and node$2
10	BRS	L10	0	((variable$2) same BDD) and (first adj storage) and (second adj storage) and sift$3
11	BRS	L11	2	((variable$2) same BDD) and (first adj storage) and (second adj storage) and processor
12	BRS	L12	1669	(proving same properties)
13	BRS	L13	19	(proving same properties) same hardware
14	BRS	L14	0	((proving same properties) same hardware) and (first adj storage) and bits
15	BRS	L15	0	((proving same properties) same hardware) and (first adj storage adj circuit$3)
16	BRS	L16	0	((proving same properties) same hardware) and (interal same signal$2)
17	BRS	L17	7	((proving same properties) same hardware) and (internal same signal$2)
18	BRS	L19	0	((proving same properties) same hardware) and (internal same signal$2) and (function$2) and (second adj storage adj circuit$2)
19	BRS	L18	7	((proving same properties) same hardware) and (internal same signal$2) and (function$2)
20	BRS	L20	0	((proving same properties) same hardware) and (internal same signal$2) and (function$2) and circuit
21	BRS	L21	7	((proving same properties) same hardware) and (internal same signal$2) and (function$2)
22	BRS	L22	23622	(processor) same (internal) same signals

	DBs	Time Stamp	Comments	Error Definition
1	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	2006/03/27 13:32		
2	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	2006/03/27 13:32		
3	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	2006/03/27 14:11		
4	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	2006/03/27 13:36		
5	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	2006/03/27 14:38		
6	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	2006/03/27 14:41		
7	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	2006/03/27 15:01		
8	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	2006/03/27 16:06		
9	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	2006/03/27 15:04		
10	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	2006/03/27 15:05		
11	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	2006/03/27 15:07		
12	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	2006/03/27 16:07		
13	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	2006/03/27 16:07		
14	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	2006/03/27 16:07		
15	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	2006/03/27 16:08		
16	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	2006/03/27 16:08		
17	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	2006/03/27 16:08		
18	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	2006/03/27 16:09		
19	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	2006/03/27 16:10		
20	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	2006/03/27 16:10		
21	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	2006/03/27 16:27		
22	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	2006/03/27 16:47		
23	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	2006/03/27 16:47		

23	BRS	L23	3	((processor) same (internal) same signals) and (BDD) and tree